

# Geometric-Algebra Quantum-Like Algorithms: Simon's Algorithm

Tomasz Magulski\*, Łukasz Orłowski†

*Department of Theoretical Physics and Quantum Informatics  
Gdańsk University of Technology, 80-952 Gdańsk, Poland*

## Abstract

This is continuation of the approach to performing quantum algorithms using geometric structures which was presented by D. Aerts and M. Czachor in [1]. We solve the Simon's problem which, next to the Shor's algorithm, is a representative of quantum hidden subgroup class. We also highlight some advantages resulting from the fact that no quantum mechanics is involved.

## 1 Problem

Consider a function

$$f : \{0, 1\}^n \rightarrow \{0, 1\}^m, \text{ where } m \geq n - 1 \quad (1)$$

we know that  $f$  is 1-to-1, or there exists  $s$  such that

$$\exists_{s \in \{0,1\}^n \setminus \{0^{(n)}\}} \forall_{x \neq y} f(x) = f(y) \iff x = y \oplus s, \quad (2)$$

where  $\oplus$  denotes componentwise addition mod 2, i.e.  $n$ -dimensional XOR.

The problem is to determine which of these conditions is satisfied by  $f$ , and, in second case, to find  $s$ .

## 2 Quantum Solution

Let us use two quantum registers, which consist of  $n$  and  $m$  qbits respectively. We start with  $n + m$  0s.

$$|\phi_0\rangle = |\underbrace{0 \dots 0}_n\rangle |\underbrace{0 \dots 0}_m\rangle \quad (3)$$

Quantum algorithm solves the problem using 3 quantum gates. At first the Hadamard gate acts on the first register. Note that using  $n$ -dimensional qubit space we have tensor power of  $n$  Hadamard gates.

$$|\phi_1\rangle = (H_n |0^{(n)}\rangle) |0^{(m)}\rangle = \frac{1}{\sqrt{2^n}} \left( \sum_{k=0}^{2^n-1} |k\rangle \right) |0^{(m)}\rangle \quad (4)$$

Second gate's action depends on function  $f$ . The gate fills the second register with function  $f$  values, using first register as a set of arguments.

$$|\phi_2\rangle = U_f |\phi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k, f(k)\rangle \quad (5)$$

---

\*email: magul@rudy.mif.pg.gda.pl

†email: shangri@rudy.mif.pg.gda.pl

Finally the Hadamard gate acts on the first register again.

$$|\phi_3\rangle = \frac{1}{\sqrt{2^n}} \left( \sum_{k=0}^{2^n-1} (H_n |k\rangle) |f(k)\rangle \right) = \frac{1}{2^n} \sum_{i,k=0}^{2^n-1} (-1)^{\sigma(i,k)} |i, f(k)\rangle \quad (6)$$

where  $\sigma$  is a dot product of binary representations of  $i$  and  $k$  over  $\{0, 1\}^n$  space, i.e.

$$\sigma(i, j) = \sum_{k=1}^n i_k^{(2)} j_k^{(2)} \quad (7)$$

Now we measure the first register.

Remember that if  $f$  has the mask  $s$  then  $f(x) = f(x \oplus s)$ , which means that our amplitude is a superposition of two amplitudes generated from  $|k, f(k)\rangle$  and  $|k \oplus s, f(k)\rangle$  in  $|\phi_2\rangle$ .

$$\alpha_{i,f(k)} = \frac{1}{2^n} \left( (-1)^{\sigma(i,k)} + (-1)^{\sigma(i,k \oplus s)} \right) \quad (8)$$

Employing

$$\sigma(i, k \oplus s) = \sigma(i, k) + \sigma(i, s) \pmod{2} \quad (9)$$

then

$$\alpha_{i,f(k)} = \frac{1}{2^n} (-1)^{\sigma(i,k)} \left( 1 + (-1)^{\sigma(i,s)} \right) \quad (10)$$

Now if  $\sigma(i, s) = 1 \pmod{2}$  then both the amplitude and probability of getting such a state is 0. This implies that every result of measurement satisfies

$$i_0 s_0 + i_1 s_1 + \cdots + i_{n-1} s_{n-1} = 0 \pmod{2} \quad (11)$$

So in order to determine whether the mask  $s$  exists the procedure has to be repeated until  $n - 1$  linearly independent states  $i$  are found.

Then the following system of equations has to be solved

$$\begin{cases} i_0^{(1)} s_0^* + i_1^{(1)} s_1^* + \cdots + i_{n-1}^{(1)} s_{n-1}^* & = 0 \pmod{2} \\ i_0^{(2)} s_0^* + i_1^{(2)} s_1^* + \cdots + i_{n-1}^{(2)} s_{n-1}^* & = 0 \pmod{2} \\ \vdots & \vdots \\ i_0^{(n-1)} s_0^* + i_1^{(n-1)} s_1^* + \cdots + i_{n-1}^{(n-1)} s_{n-1}^* & = 0 \pmod{2} \end{cases} \quad (12)$$

There are two possible solutions of this system -  $0^{(n)}$  and  $s^*$ .

Now recalling the definition we know that  $f$  is 1-to-1 or, there exists  $s$  which satisfies (2). In the second case the non-trivial solution of (12) is the mask we are looking for. In the first one it is easy to show that  $s^*$  is just a random sequence. The easiest way to determine which of these conditions are satisfied by  $f$  is to check if  $f(x) = f(x \oplus s^*)$  where  $x \in \{0, 1\}^n$ .

Originally Simon's problem was defined and solved in [2].

### 3 GA formulation

GA formulation of the problem is based on the binary parametrization [3]. Consider  $(n+m)$ -dimensional space with orthonormal basis  $\{e_1 \dots e_{n+m}\}$  and its associated GA. Initial state is analogous to  $|\phi_0\rangle$ :

$$\underbrace{e_0 \dots 0}_{n+m} \quad (13)$$

Let us use a multivector

$$E_n = \sum_{A_1, A_2, \dots, A_n=0}^1 e_{A_1 A_2 \dots A_n} \underbrace{0 \dots 0}_m \quad (14)$$

then

$$E_n \underbrace{e_0 \dots 0}_{n+m} = \sum_{A_1, A_2, \dots, A_n=0}^1 e_{A_1 A_2 \dots A_n} \underbrace{0 \dots 0}_m \quad (15)$$

This step is equivalent to the first Hadamard gate in quantum algorithm.  $U_f$  is an operation which performs as follows

$$U_f E_n \underbrace{e_0 \dots 0}_{n+m} = \sum_{A_1, A_2, \dots, A_n=0}^1 e_{A_1 A_2 \dots A_n} f_1(A_1 A_2 \dots A_n) \dots f_m(A_1 A_2 \dots A_n) \quad (16)$$

Note that  $f_i(A_1 A_2 \dots A_n)$  is the  $i$ -th function  $f$  value.

Let us consider the reverse of  $E_n$

$$F_n = \sum_{B_1, B_2, \dots, B_n=0}^1 e_{B_1 B_2 \dots B_n}^\dagger \underbrace{0 \dots 0}_m \quad (17)$$

Employing [3]

$$e_{X_1 X_2 \dots X_n} e_{Y_1 Y_2 \dots Y_n} = (-1)^{\sum_{i < j} Y_i X_j} e_{(X_1 \dots X_n) \oplus (Y_1 \dots Y_n)} \quad (18)$$

and

$$e_{X_1 X_2 \dots X_n}^\dagger e_{X_1 X_2 \dots X_n} = 1 \quad (19)$$

we find

$$e_{X_1 X_2 \dots X_n}^\dagger = (-1)^{\sum_{i < j} X_i X_j} e_{X_1 X_2 \dots X_n} \quad (20)$$

Therefore

$$F_n = \sum_{B_1, B_2, \dots, B_n=0}^1 (-1)^{\sum_{i < j} B_i B_j} e_{B_1 B_2 \dots B_n} \underbrace{0 \dots 0}_m \quad (21)$$

$F_n$  performs as follows

$$F_n U_f E_n e_{\underbrace{0 \dots 0}_{n+m}} = \sum_{B_1, B_2, \dots, B_n=0}^1 \left( (-1)^{\sum_{i<j} B_i B_j} e_{B_1 B_2 \dots B_n} \underbrace{0 \dots 0}_m \right) \sum_{A_1, A_2, \dots, A_n=0}^1 e_{A_1 A_2 \dots A_n} f_1(A_1 A_2 \dots A_n) \dots f_m(A_1 A_2 \dots A_n) \quad (22)$$

$$= \sum_{A_1, \dots, A_n, B_1, \dots, B_n=0}^1 (-1)^{\sum_{i<j} B_i B_j} e_{B_1 \dots B_n} \underbrace{0 \dots 0}_m e_{A_1 \dots A_n} f_1(A_1 \dots A_n) \dots f_m(A_1 \dots A_n) \quad (23)$$

$$= \sum_{A_1, \dots, A_n, B_1, \dots, B_n=0}^1 (-1)^{\sum_{i<j} B_i B_j + \sum_{i<j} A_i B_j} e_{(B_1 \dots B_n) \oplus (A_1 \dots A_n)} f_1(A_1 \dots A_n) \dots f_m(A_1 \dots A_n) \quad (24)$$

Now let us focus on the case in which  $f$  has the mask  $s$ . Think how the values standing next to each blade of the state (24) look like. Note that for each  $f$  value the inverse image of  $f$  is a two-element set. This implies that for all  $(A_n) \in \{0, 1\}^n$ , in a state  $U_f E_n e_{0 \dots 0}$  there are exactly two blades with  $f((A_n))$  values on the last  $m$  bits:

$$e_{A_1 \dots A_n} f_1(A_1 \dots A_n) \dots f_m(A_1 \dots A_n) \quad (25)$$

and

$$e_{(A_1 \dots A_n) \oplus (S_1 \dots S_n)} f_1(A_1 \dots A_n) \dots f_m(A_1 \dots A_n) \quad (26)$$

where  $(A_n)$  is arbitrary and  $(S_n)$  is the mask we want to find.

Note that for all  $(A_n) \in \{0, 1\}^n$  there is a sequence in  $\{0, 1\}^n$  which by means of XOR operation can create an arbitrary sequence in  $\{0, 1\}^n$ . Therefore, since  $F_n$  contains all possible blades with 0s on the last  $m$  bits, its action on  $U_f E_n e_{0 \dots 0}$  gives us (from blades (25) and (26)) two components of coefficient standing next to the blade:

$$e_{X_1 \dots X_n} f_1(A_1 \dots A_n) \dots f_m(A_1 \dots A_n) \quad (27)$$

where  $(X_n)$  is arbitrary.

Let us now consider  $(X_n)$  and  $(A_n)$ . What is the coefficient standing next to (27)? It would be convenient to denote it by  $\alpha_{X, f(A)}$ .

The first component resulting from (25) is

$$(-1)^{\sum_{i<j} B_i B_j + \sum_{i<j} A_i B_j} = (-1)^{\sum_{i<j} (B_i + A_i) B_j} \quad (28)$$

where  $(B_n)$  satisfies  $(X_n) = (A_n) \oplus (B_n)$ . The second component resulting from (26) is

$$(-1)^{\sum_{i<j} B_i^\oplus B_j^\oplus + \sum_{i<j} (A_i \oplus S_i) B_j^\oplus} = (-1)^{\sum_{i<j} (B_i^\oplus + (A_i \oplus S_i)) B_j^\oplus} \quad (29)$$

where  $(B_n^\oplus)$  satisfies  $(X_n) = ((A_n) \oplus (S_n)) \oplus (B_n^\oplus)$ . Adding the values from (28) and (29) we get the coefficient

$$\alpha_{X, f(A)} = (-1)^{\sum_{i<j} (B_i + A_i) B_j} + (-1)^{\sum_{i<j} (B_i^\oplus + (A_i \oplus S_i)) B_j^\oplus} \quad (30)$$

which corresponds to a blade in a state  $F_n U_f E_n e_0 \dots 0$ . The coefficients of this blade's binary parametrization carry no information about  $(A_n)$  which means we also have no information about  $(B_n)$ ,  $(B_n^\oplus)$ , let alone  $(S_n)$ . However there are some conclusions to be drawn. Note that for all  $i$

$$X_i = A_i \oplus B_i = (A_i \oplus S_i) \oplus B_i^\oplus \quad (31)$$

which implies

$$B_i^\oplus = A_i \oplus B_i \oplus A_i \oplus S_i \quad (32)$$

$$B_i^\oplus = B_i \oplus S_i \quad (33)$$

Employing this in (30) we have

$$\alpha_{X,f(A)} = (-1)^{\sum_{i<j} (B_i+A_i)B_j} + (-1)^{\sum_{i<j} ((B_i \oplus S_i) + (A_i \oplus S_i))(B_j \oplus S_j)} \quad (34)$$

$f(n) = (-1)^n$  is periodic with period 2 which allows us to switch some pluses for XOR operations and the other way round

$$\begin{aligned} \alpha_{X,f(A)} &= (-1)^{\sum_{i<j} (B_i+A_i)B_j} + (-1)^{\sum_{i<j} (B_i \oplus S_i \oplus A_i \oplus S_i)(B_j \oplus S_j)} \\ &= (-1)^{\sum_{i<j} (B_i+A_i)B_j} + (-1)^{\sum_{i<j} (B_i \oplus A_i)(B_j \oplus S_j)} \\ &= (-1)^{\sum_{i<j} (B_i+A_i)B_j} + (-1)^{\sum_{i<j} (B_i+A_i)(B_j+S_j)} \\ &= (-1)^{\sum_{i<j} (B_i+A_i)B_j} + (-1)^{\sum_{i<j} (B_i+A_i)B_j + \sum_{i<j} (B_i+A_i)S_j} \\ &= (-1)^{\sum_{i<j} (B_i+A_i)B_j} \left( 1 + (-1)^{\sum_{i<j} (B_i+A_i)S_j} \right) \end{aligned} \quad (35)$$

Now we can see that whether the particular blade occurs in the final multivector or zeros itself depends on the value of the exponent (expression in brackets).

Note that unlike the quantum solution where each measurement gives us only one of the basic states from the superposition, in GA formulation the observation has no influence on the multivector so we can simply observe every particular amplitude of the blade we are interested in.

We have reached the point where we are able to determine whether  $f$  is 1-to-1 or there exists the mask  $s$ . According to (35)  $f$  has the mask if absolute values of amplitudes from the multivector  $F_n U_f E_n e_0 \dots 0$  equal 2. On the other hand, if  $f$  is 1-to-1, for all blades in (15) the last  $m$  bits in (16) are unique. This implies that absolute values of all amplitudes in (22) equal 1.

Supposing that the mask  $s$  exists let us focus on finding it. From (35) we have

$$\begin{cases} \sum_{i<j} (B_i + A_i) S_j = 0 \pmod{2} & \text{for non-zero-blades} \\ \sum_{i<j} (B_i + A_i) S_j = 1 \pmod{2} & \text{for zero-blades} \end{cases} \quad (36)$$

and because of modulo 2 operation

$$\begin{cases} \sum_{i<j} (B_i \oplus A_i) S_j = 0 \pmod{2} & \text{for non-zero-blades} \\ \sum_{i<j} (B_i \oplus A_i) S_j = 1 \pmod{2} & \text{for zero-blades} \end{cases} \quad (37)$$

Employing (31)

$$\begin{cases} \sum_{i<j} X_i S_j = 0 \pmod{2} & \text{for non-zero-blades} \\ \sum_{i<j} X_i S_j = 1 \pmod{2} & \text{for zero-blades} \end{cases} \quad (38)$$

In terms of Simon's problem we examine the amplitudes of blades which are in the form of

$$\underbrace{e0 \dots 0}_{k} 11 \underbrace{0 \dots 0}_{n+m-k-2} \quad (39)$$

where  $k \in \langle 0, n-2 \rangle$ . Returning to our system we get

$$\begin{cases} \sum_{i < j < k+2} X_i S_j + \sum_{i < k+2} X_i S_{k+2} + \sum_{i < j, j > k+2} X_i S_j = 0 \pmod{2} & \text{for non-zero-blades} \\ \sum_{i < j < k+2} X_i S_j + \sum_{i < k+2} X_i S_{k+2} + \sum_{i < j, j > k+2} X_i S_j = 1 \pmod{2} & \text{for zero-blades} \end{cases} \quad (40)$$

Employing (39)

$$\begin{cases} \sum_{1 < j < k+2} 0S_j + 1S_{k+2} + \sum_{j > k+2} 2S_j = 0 \pmod{2} & \text{for non-zero-blades} \\ \sum_{1 < j < k+2} 0S_j + 1S_{k+2} + \sum_{j > k+2} 2S_j = 1 \pmod{2} & \text{for zero-blades} \end{cases} \quad (41)$$

Again because of the modulo 2 operation

$$\begin{cases} S_{k+2} = 0 \pmod{2} & \text{for non-zero-blades} \\ S_{k+2} = 1 \pmod{2} & \text{for zero-blades} \end{cases} \quad (42)$$

Now we can determine all the bits of our mask from  $S_2$  to  $S_n$ . To find  $S_1$  we have to check two possible masks  $s$ :  $(0S_2 \cdots S_n)$  and  $(1S_2 \cdots S_n)$ .

## 4 Explicit Examples

In this section we want to show explicitly how GA formulation works. We present two examples.

Let us consider the following 1-to-1 function

$X$	0	0	1	1
	0	1	0	1
$f(X)$	1	0	1	0
	0	0	1	1

Table 1: 1-to-1 function

and perform Simon's algorithm

$$e_{0000} \quad (43)$$

$$E_n e_{0000} = e_{0000} + e_{0100} + e_{1000} + e_{1100} \quad (44)$$

$$U_f E_n e_{0000} = e_{0010} + e_{0100} + e_{1011} + e_{1101} \quad (45)$$

$$\begin{aligned} F_n U_f E_n e_{0000} &= e_{0000} + e_{0001} + e_{0010} + e_{0011} + e_{0100} + e_{0101} + e_{0110} + e_{0111} \\ &\quad - e_{1000} - e_{1001} + e_{1010} + e_{1011} + e_{1100} + e_{1101} - e_{1110} - e_{1111} \end{aligned} \quad (46)$$

We can see that the absolute value of every amplitude in the multivector is 1 which proves that  $f$  is 1-to-1.

Now let us define the function which has a mask

$X$	0	0	0	0	1	1	1	1
	0	0	1	1	0	0	1	1
	0	1	0	1	0	1	0	1
$f(X)$	1	0	1	0	1	0	1	0
	1	0	1	0	0	1	0	1

Table 2: Function with mask  $s = (010)$

and perform the algorithm

$$e_{000000} \quad (47)$$

$$E_n e_{000000} = e_{000000} + e_{00100} + e_{01000} + e_{01100} + e_{10000} + e_{10100} + e_{11000} + e_{11100} \quad (48)$$

$$U_f E_n e_{000000} = e_{00011} + e_{00100} + e_{01011} + e_{0110} + e_{10010} + e_{10101} + e_{11010} + e_{11100} \quad (49)$$

$$\begin{aligned} F_n U_f E_n e_{000000} &= 2(e_{000000} + e_{00001} + e_{00010} + e_{00011} + e_{00100} + e_{00101} + e_{00110} + e_{00111} \\ &\quad - e_{01000} - e_{01001} + e_{01010} + e_{01011} + e_{01100} + e_{01101} - e_{01110} - e_{01111}) \end{aligned} \quad (50)$$

In this case the absolute value of every amplitude in the multivector equals 2 and therefore the non-trivial mask  $s$  exists.

In Sec. 3 we showed how to determine our mask  $s$  using blades in the form of (39).

Let us illustrate how it works with the help of the following table

blade's binary parametrization	0	0	0	0	1	1	1	1	0	1
	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	0	0	0	0	1	0
	0	0	1	1	0	0	1	1	X	X
	0	1	0	1	0	1	0	1	X	X
absolute value of an amplitude	2	2	2	2	0	0	0	0	2	0

Table 3: Blade's amplitudes

Thanks to (42) we have

$$\begin{cases} S_{0+2} = S_2 = 1 & \text{because the amplitude of } e_{110XX} \text{ equals 0} \\ S_{1+2} = S_3 = 0 & \text{because the amplitude of } e_{011XX} \text{ equals 2} \end{cases} \quad (51)$$

and therefore  $s = (X10)$ . To determine the first bit we need to evaluate  $f(000)$  and  $f(010)$ . In our example  $f(000) = f(010)$  so  $S_1 = 0$ . In this example the algorithm has proved that the mask  $s$  exists and equals (010).

## 5 Cartan's representation

Let us use the matrix algebra known as Cartan's representation of 1-blade in GA [6]:

$$e_{2k-1} = \underbrace{\sigma_1 \otimes \cdots \otimes \sigma_1}_{n-k} \otimes \sigma_3 \otimes \underbrace{\mathbb{1} \otimes \cdots \otimes \mathbb{1}}_{k-1} \quad (52)$$

$$e_{2k} = \underbrace{\sigma_1 \otimes \cdots \otimes \sigma_1}_{n-k} \otimes \sigma_2 \otimes \underbrace{\mathbb{1} \otimes \cdots \otimes \mathbb{1}}_{k-1} \quad (53)$$

and obviously the scalar representation is

$$e_0 = \underbrace{\mathbb{1} \otimes \cdots \otimes \mathbb{1}}_n \quad (54)$$

where  $\sigma_1, \sigma_2, \sigma_3$  are the Pauli matrices and  $\mathbb{1}$  denotes  $2 \times 2$  unity matrix.  
An arbitrary blade can be represented by the adequate product of (52) and (53).  
We know that

$$\text{Tr } \sigma_1 = \text{Tr } \sigma_2 = \text{Tr } \sigma_3 = 0 \quad (55)$$

and

$$\text{Tr} \left( \bigotimes_{i=1}^n A_i \right) = \prod_{i=1}^n \text{Tr } A_i \quad (56)$$

$$\text{Tr} \left( \sum_{i=1}^n A_i \right) = \sum_{i=1}^n \text{Tr } A_i \quad (57)$$

Therefore for the scalar

$$\text{Tr } e_0 = (\text{Tr } \mathbb{1})^n = 2^n \quad (58)$$

and for 1-blades

$$\text{Tr } e_{2k-1} = (\text{Tr } \sigma_1)^{n-k} \text{Tr } \sigma_3 (\text{Tr } \mathbb{1})^{k-1} = 0 \quad (59)$$

$$\text{Tr } e_{2k} = (\text{Tr } \sigma_1)^{n-k} \text{Tr } \sigma_2 (\text{Tr } \mathbb{1})^{k-1} = 0 \quad (60)$$

It is easy to show that

$$\begin{cases} \text{Tr } (e_{A_1 \dots A_n} e_{B_1 \dots B_n}) = 0 & \text{if } \exists_i A_i \neq B_i \\ \text{Tr } (e_{A_1 \dots A_n}^2) = (-1)^{\frac{\sum_{i=1}^n A_i (\sum_{i=1}^n A_i - 1)}{2}} 2^n \end{cases} \quad (61)$$

So having a multivector

$$X = \sum_{A_1, \dots, A_n=0}^1 X_{A_1 \dots A_n} e_{A_1 \dots A_n} \quad (62)$$

we can compute the coefficients by

$$X_{A_1 \dots A_n} = \frac{(-1)^{\frac{\sum_{i=1}^n A_i (\sum_{i=1}^n A_i - 1)}{2}}}{2^n} \text{Tr } (e_{A_1 \dots A_n} X) \quad (63)$$



Let us consider the problem for the 1-to-1 function from the previous section (Table 1). The initial state  $e_{0000}$  has the following representation:

[illegible]

Note that

[illegible]

In general the second gate is not a multivector so it does not have Cartan's representation and therefore only the result of  $U_f E_n e_{0000}$  can be represented:

[illegible]

Then

[illegible]

and its action on  $U_f E_n e_{0000}$

$$\begin{pmatrix} 1-i & 1-i & -1-i & 1+i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-i & -1+i & -1-i & -1-i \\ -1-i & 1+i & 1-i & 1-i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1+i & 1+i & -1+i & 1-i \\ 1-i & 1-i & 1+i & -1-i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+i & 1-i & -1-i & -1-i \\ 1+i & -1-i & 1-i & 1-i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+i & 1+i & 1-i & -1+i \\ 0 & 0 & 0 & 0 & 1-i & 1-i & -1-i & 1+i & 1-i & -1+i & -1-i & -1-i & -1-i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1-i & 1+i & 1-i & 1-i & 1+i & 1+i & -1+i & -1+i & 1-i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-i & 1-i & 1+i & -1-i & -1-i & 1-i & -1-i & -1-i & -1-i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+i & -1-i & 1-i & 1-i & 1+i & 1+i & 1-i & 1-i & -1+i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-i & -1+i & -1-i & -1-i & 1-i & 1-i & -1-i & 1-i & 1-i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+i & 1+i & -1+i & 1-i & -1-i & 1-i & -1-i & 1-i & 1-i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1+i & 1-i & -1-i & -1-i & -1-i & 1-i & 1-i & 1-i & -1-i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+i & 1+i & 1-i & -1+i & 1+i & -1-i & 1-i & 1-i & 1-i & 0 & 0 & 0 & 0 \\ 1-i & -1+i & -1-i & -1-i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-i & 1-i & -1-i & 1+i \\ 1+i & 1+i & -1+i & 1-i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1-i & 1+i & 1-i & 1-i \\ -1+i & 1-i & -1-i & -1-i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-i & 1-i & 1+i & -1-i \\ 1+i & 1+i & 1-i & -1+i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+i & -1-i & 1-i & -1-i \end{pmatrix} \quad (68)$$

Now using (63) we can find the amplitudes (Table 4)

$A_1$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
$A_2$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
$A_3$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
$A_4$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$\text{Tr}(e_{A_1 A_2 A_3 A_4} F_n U_f E_n e_{0000})$	16	16	16	-16	16	-16	-16	-16	-16	16	-16	-16	-16	-16	16	-16
amplitude of $e_{A_1 A_2 A_3 A_4}$	1	1	1	1	1	1	1	1	-1	-1	1	1	1	1	-1	-1

Table 4: Traces of matrices

which give us the following multivector:

$$\begin{aligned} F_n U_f E_n e_{0000} &= e_{0000} + e_{0001} + e_{0010} + e_{0011} + e_{0100} + e_{0101} + e_{0110} + e_{0111} \\ &\quad - e_{1000} - e_{1001} + e_{1010} + e_{1011} + e_{1100} + e_{1101} - e_{1110} - e_{1111} \end{aligned} \quad (69)$$

Note that it is the same as (46).

Let us consider the function with non-trivial mask  $s = (10)$ :

$X$	0	0	1	1
	0	1	0	1
$f(X)$	1	0	1	0
	0	1	0	1

Table 5: Function with mask  $s = (10)$

For some technical reason it is less dimensional function than (Table 2) from the previous section.

Note that  $e_{0000}$ ,  $E_n = E_n e_{0000}$  and  $F_n$  are exactly the same as (64), (65) and (67) respectively. We have

$$U_f E_n e_{0000} = \begin{pmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (70)$$

and

$$F_n U_f E_n e_{0000} = \begin{pmatrix} -2i & 2 & -2 & 2i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2i & -2i & -2 \\ -2 & 2i & -2i & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2i & 2 & -2 & -2i \\ 2 & -2i & 2i & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2i & 2 & -2 & -2i \\ 2i & -2 & 2 & -2i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2i & -2i & -2 \\ 0 & 0 & 0 & 0 & -2i & 2 & -2 & 2i & 2 & 2i & -2i & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 2i & -2i & 2 & 2i & 2 & -2 & -2i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2i & 2i & -2 & 2i & 2 & -2 & -2i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2i & -2 & 2 & -2i & 2 & 2i & -2i & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2i & -2i & -2 & -2i & 2 & -2 & 2i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2i & 2 & -2 & -2i & -2 & 2i & -2i & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2i & 2 & -2 & -2i & 2 & -2i & 2i & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2i & -2i & -2 & 2i & -2 & 2 & -2i & 0 & 0 & 0 & 0 \\ 2 & 2i & -2i & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2i & 2 & -2 & 2i \\ 2i & 2 & -2 & -2i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2i & -2i & 2 \\ 2i & 2 & -2 & -2i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2i & 2i & -2 \\ 2 & 2i & -2i & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2i & -2 & 2 & -2i \end{pmatrix} \quad (71)$$

Again using (63) we can find the amplitudes (Table 6)

$A_1$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
$A_2$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
$A_3$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
$A_4$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$\text{Tr}(e_{A_1 A_2 A_3 A_4} F_n U_f E_n e_{0000})$	0	32	32	0	0	-32	-32	0	0	32	-32	0	0	-32	32	0
amplitude of $e_{A_1 A_2 A_3 A_4}$	0	2	2	0	0	2	2	0	0	-2	2	0	0	2	-2	0

Table 6: Traces of matrices

which give us the following multivector:

$$F_n U_f E_n e_{0000} = 2(e_{0001} + e_{0010} + e_{0101} + e_{0110} - e_{1001} + e_{1010} + e_{1101} - e_{1110}) \quad (72)$$

Performing (39-42) as we did in Sec. 4 using the function (Table 2) we find the mask  $s = (10)$ .

## 6 Acknowledgement

We are deeply indebted to Marek Czachor for stimulating suggestions, help and encouragement.

## References

- [1] D. Aerts, M. Czachor, "Cartoon computation: Quantum-like algorithms without quantum mechanics", J. Phys. A: Math. Theor. 40, F259-F266 (2007)
- [2] D.R. Simon, "On the Power of Quantum Computation", SIAM J. Comput. 26:1474-1483 (1997)
- [3] D. Aerts, M. Czachor, and B. De Moor, "On Geometric Algebra representations of Binary Spatter Codes", cs.AI/0610075 (2006)
- [4] D. Aerts, M. Czachor, "Notes on Geometric-Algebra Quantum-Like Algorithms", quantum-ph/0610178 (2006)
- [5] Jaap Suter, "Geometric Algebra Primer", <http://lomont.org/Math/GeometricAlgebra/GeometricAlgebraPrimer-Suter-2003.pdf> (2003)
- [6] P. Budinich, A. Trautman, "The Spinorial Chessboard", Springer, New York (1988).